Evolutionary Multi-agent Diffusion Approach to Optimization

Kwok Ching Tsui and Jiming Liu

Department of Computer Science

Hong Kong Baptist University

Kowloon Tong, Kowloon, Hong Kong

email: {tusikc,jiming}@comp.hkbu.edu.hk

December 5, 2001

Abstract

This article proposes a novel multi-agent approach to optimization inspired by diffusion in nature called *Evolutionary Multi-agent Diffusion* (EMD). Each agent in EMD makes the decision to diffuse based on the information shared between its parent and its siblings. The behvior of EMD is analyzed and its relation to similar search algorithms is discussed.

Keywords: evolutionary diffusion, mutil-agent system, emergent behavior, autonomy oriented computation, optimization

Abbreviated title: Evolutionary Multi-agent Diffusion

Contents

List of Figures

1 Introduction

Consider a function $F(x)$ and $x = \{x_1, x_2, ..., x_n\}^T$ is a n-dimensional vector representing the parameters of function F. The optimal solution is represented by $F(x^*)$ such that

$$F(x^*) < F(x) \quad \forall x \tag{1}$$

The search for x^* can be viewed as the minimization of function F. Turning the sign in equation 1 around makes the search for x^* a maximization task. They can collectively be called global optimization tasks [13].

Many algorithms has been developed over the years to tackle the challenging task of global optimization [8, 7, 11]. There are several aspects to the challenge any new search algorithm will face. Firstly, the landscape of the function to be optimized in unknown. Unimodal functions can be monotonic in nature and the search is easy once the downhill 'direction' is found. On the other hand, multimodal functions have many suboptimal solutions, i.e. places where the objective function value is lower than the neighborhood areas but is not the optimal solution. A search algorithm is likely to trapped in these suboptimal solutions.

Secondly, the dimensionality of the function presents problems to a search algorithm. When changes are made to the value of one or more parameters, it is not immediately known how much contribution each of those changes are to the change in the function value (or the absence of change, for the matter).

A search algorithm is a carefully formulated systematic approach to locate the optimal solution. It needs to make decisions regarding modifications to parameter set of a candidate solution and to interpret the change in the function value before and after the change. These rules are captured in a policy. A *stochastic search algorithm* employs a policy that do not make the same decision all the time. They have been proved to be efficient in many numerical optimization tasks. Simulated annealing is a classic example of such search algorithm that maintain a single candidate solution at all times [10].

Population-based search algorithms employ many candidate solutions at all times. Depending on the policy of individual algorithms, candidate solutions are mixed to create new ones or simply modified. The new and old candidates together are then selected according to how they perform with respect to the function. A fixed size population is normally used. Evolutionary algorithms [1], particle swarm [9], cultural algorithm [12] and ant colony optimization [5] are population-based search algorithms.

This article proposes a new population-based stochastic search algorithm called *evolutionary diffusion optimization* (EMD) inspired by diffusion in nature. Unlike the population-based algorithms mentioned above, the population size in EMD is not fixed because it is very 'lenient' towards poor performers in the population. Moreover, individual candidate solutions are changed based on information obtained in its proximity. The right to reproduce is also granted based on local competition.

2 Diffusion in Nature

A drop of red ink in a glass of water will eventually stain the whole glass red. Likewise, people move around the globe.

Things or people diffuse, move or migrate from place to place for different reasons. The red ink is able to diffuse throughout the whole glass of water because there is a difference in the concentration of the ink molecules. This concentration gradient acts as the driving force to draw the ink molecules away from their initial positions. The counteracting force is the viscosity of the water, which tends to pull the ink molecules from diffusing. The balance of these two forces results in the movement of the ink molecules [4].

The study of human geography categorized reasons for human migration into push factors and pull factors [2, 3]. Push factors relate to undesirable conditions such as poor living conditions, lack of job and overcrowding. Pull factors are those positive factors that attracts people to relocate, such as jobs and better living conditions. Two forms of migration can also be identified. Step migration refers to a series of local movements, such as moving from village to town, then to city. Chain migration refers to a more drastic change beyond the local region and is usually assisted by people who have already emigrated. The availability of information seems to be an important factor that help people to decide when and where to migrate.

While diffusion in chemical world is analogous to migration in the human world, there are more commonplace between them than just this. Some kind of driving force is the most important factor in facilitating any motion. This is an important lesson to learn for function optimization, but a difficult one. It is obvious that once the landscape of the function to be optimized is known, finding the optimal solution becomes trivial. The question is how to capture the 'trend'.

In addition, 'Knowledgeable' people can help others to decide whether to move or not. In essence, whoever has captured the 'trend' of the search space can help others to make 'better' moves. The chance of finding a solution can possibly be enhanced by making small 'step' migrations. The following section describes the evolutionary multi-agent diffusion approach to optimization that has facility to learn the 'trend' of the search space, while forming chains of information flow to facilitate small step exploration of the search space.

3 The Algorithm

3.1 Basic Data Structure

3.1.1 Object Vector

An agent in EMD represents a candidate solution to the optimization problem in hand. Therefore, each agent carries all the object variables of the function to be optimized and are tuned at the same time by EMD.

3.1.2 Probability Matrix

Each agent has a probability matrix associated with it. The matrix contains the likelihood estimate of success with respect to direction of motion. Specifically, the probability matrix is an $m \times n$ matrix representing the m object variables to be optimized and n possible steps of motions, i.e. $(n-1)/2$ steps towards the upper bound and the lower bound respectively, and remain stationery. At the beginning of the search, all entries are initialized to the $1/n$, which means it is equally likely to make any of the possible moves. The probability matrix will be continuously updated as a result of the local behavior of an agent. As the probabilities for various steps are updated, EMD begins to differentiate between 'good' and 'bad' directions of move with respect to the initial position of the ancestor. The probability matrix, therefore, carries some kind of 'heritage' throughout the family line of those agents that were present when the search begins.

Parents and their offspring share the same probability matrix. It is only when an agent becomes a parent then a probability matrix will be created for it, which is an exact copy of the parent's updated matrix. Sharing the probability matrix between parents and siblings enables agents from the same family to learn from each other's successes as well as failures. Details of this feedback sharing feature is discussed in more detail below.

3.1.3 Fitness

Fitness in EMD measures an agent's degree of success in the course of the search from the optimal solution. It is also used as a basis to determine various local behaviour. For simplicity, the objective function value is used as fitness in EMD if the task is minimization. The reciprocal of the function value can be used as fitness if EMD is used in a maximization task.

3.2 Local Behaviours

3.2.1 Diffuse

Agents in EMD explore uncharted locations in the solution space by diffusion. In the majority of time, the agents will behave rationally and choice a new location according to the probability distribution given by the probability matrix. These moves will result in a fitness higher than the parent as well as lower. This is because the probability matrix is only accumulation of experience of the agent's predecessors and is not the ground truth. As an agent becomes older and still have not located a 'better' place, it will probabilistically decides to act wild and take a random walk according to the following equation:

$$P_{random\_walk} = e^{-\frac{lifespan - age}{\alpha}} \tag{2}$$

where $0 \leq P_{random\_walk} \leq 1$ for $1 \leq age \leq lifespan$, and α is a scaling factor that decides the degree to which random walk is to be exercised. A restriction on diffusion is that this process must result in a position different from the agent's position before taking the move. If the agent remains stationery after diffusion, the process will be repeated. In more detail, the two types of diffusion an agent can take are described below.

a. *Rational Move* An agent modifies its object values by drawing a random number for each dimension of the object vector. Each random number is then used to choose between the set of fixed steps according to the probability matrix. When the number of steps is determined, the corresponding value is added to the entry in question in the object vector. This process is repeated until all dimensions of the object vector are covered. The updated object vector then becomes the new position of the agent in the solution space.

b. *Random Walk* when an agents decides to diffuse to a random location, it will first choose uniformly randomly a direction for each dimension, i.e towards upper bound, towards lower bound and no move. In case a move is to be made, a new value between the limit in the chosen direction and the current value is then chosen randomly. The process ends when all the dimensions of the object vector are updated.

3.2.2 Reproduce

At the end of a generation, the fitness of all (active) agents are compared with that of their parents, which have temporarily become stationary (inactive). All agents with higher fitness are allowed to reproduce. Asexual reproduction is used in EMD to explore the search space. A reproducing agent will replicate itself the number of times as dictated by the following rules. The new agents are then sent off to a new location by rational move. In the event duplicates resulted during reproduction, all of them except one are removed.

The number of offspring to breed is governed by the system-wide parameter and the following two rules:

a. *Differentiate Reproduction Quota* An agent is given the full quota to reproduce only if its fitness is 25% of the population average. Otherwise, two less than the quota is given for those in the lower 25%; one less than the quota applies to the rest.

b. *Restriction on Reproduction Quota* Further to the above differential scheme, the reproduction quota is subjected to further restriction due to overcrowding.

$$quota' = \lfloor (quota * (pop_{max} - pop_{current})/pop_{max}) \rfloor \tag{3}$$

c. *Rejuvenate* Higher performing agents whose offspring have all died will be allowed to reproduce. This mechanism is described in the next section

3.2.3 Aging

While *repduction* is the way to amplify the effect of the potentially useful agents, *aging* is used in EMD to eliminate poor performers after giving them sufficient opportunity to search and improve.

a. *Get Old* The normal lifespan of the agents is governed by a system-wide parameter. At the end of each generation, the age of all agents is increased by one. Agents whose age is greater than the lifespan limit are eliminated from the system unless the following condition (extended life) is satisfied.

b. *Extended Life* Lifespan limit of an agent is extended by one generation when it expires if its fitness is higher than the population average.

c. *Sudden Death* An unsuccessful agent will be eliminated if the agent's fitness is more than 25% of the parent's.

$$\frac{fitness}{parent'sfitness} > 1.25 \tag{4}$$

d. *Rejuvenate* An inactive agent will be allowed to spawn new offsprings if the following two conditions are satisfied:

   (i) all its offspring are dead

   (ii) its fitness is better than the current population average

   The main idea behind *rejuvenate* is that the inactive agent has been receiving positive as well as negative reinforcement signals from its offspring (described below), its probability matrix contains the latest information regarding the neighborhood landscape. It would be a waste if this potentially useful information is discarded.

e. *Permanent removal* Any inactive agent not being rejuvenated will be removed permanently from the system.

3.2.4 Feedback

All agents in EMD maintains a close link between parents and offspring. When an offspring has diffused to another location in the hyperspace, it is being evaluated. A 'good' move to a 'better' location will cause the agent in question to become a parent. This implies the information regarding directions supplied by the parent is likely to be a good set of directions to follows. Similarly, siblings need to be notified of any 'bad' moves.

a. *Positive feedback* For positive reinforcement after a successful move, which may happen after taking many moves, the probabilities relating to the positive steps, negative steps and no change, are updated if the offspring's object variable is greater than, smaller than, or equal to the corresponding object variable of the parent. The update rule is

$$p'_{id} = \frac{p_{id} + \delta}{1 + \delta}, \tag{5}$$

where the subscript d relates to the direction of move.

b. *Negative feedback* Individual agents will update the parent's probability matrix after each move, as oppose to just after becoming a parent. The unsuccessful moves will pass back negative reinforcement signals to the parent. The parent in turn will update those probabilities that relates to the ith object variable and jth step size the offspring in question has just taken using

$$p'_{ij} = p_{ij} \times (1 - \delta). \tag{6}$$

This is a finer grain update than the positive reinforcement signal as it happens after each step. Moreover, the use of a multiplicative scaling factor ensures that the probability remains bigger than zero at all times.

Note that all p' referred above are normalized using $\sum_i p'_i$ after updating.

3.3 Global Information and its Adaptation

EMD keeps several system-wide information that either controls the flow of the search or act as parameters to some of the features described above. The four paramters are:

a. *Offspring_to_breed*: Subject to other restrictions, this is the maximum number of offspring any reproducing agents is allowed to spawn at one time.

b. *Max_pop_size*: This is the upper limit of the number of agents, both active and inactive together, an EMD session can keep.

c. *Lifespan*: This duration, in generations, given to an agent to perform search

d. *Step_size*: The amount of change to an object variable. It is used in combination with the probability matrix to decide what should the actual change be. For example, if the two steps toward the upper limit were chosen based on the probability distribution in the probability matrix, a value equal to two times of *Step_size* parameter will be added to the object variable in question.

EMD has some adaptive behavior to assist the optmization process. Firstly, the *Step_size* parameter is reduced according to the golden ratio if the current best has not been renewed for half of *Lifespan* (in number of generations). The rationale behind this reduction is that the agents may be in the neighborhood of a minimum. Therefore, finer steps are required for careful exploitation. Conversely, if the population has been improving continuously for some time (in number of generations, again equal to half of *Lifespan*), the step size is increased according to the golden ratio (division).

EMD also reduces the maximum population size by 0.1% at each generation to increase the selection pressure at the search advances.

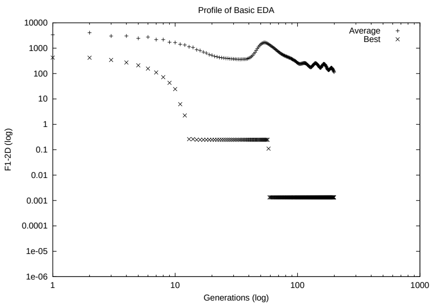
4 Discussion

Using the most basic form (two-dimensional) of equation 7 as a testbed, this section will present the behavior of the more important features of EMD outlined above.

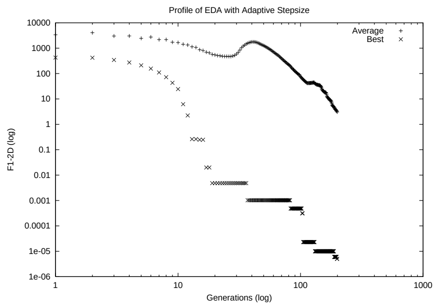
$$F_1(x) = \sum_{i=1}^{n} x_i^2 \tag{7}$$

The basic form of the evolutionary muti-agent diffusion (EMD) consists of local behaviors such as diffusion, reproduction, aging and feedback. Some advanced features such as random walk during diffusion, step size adaptation and solution rejuvenation are also available to EMD. Various combinations of the basic EMD with the advanced features are used to optimization the two-dimensional version of F1 for 200 generations. Figure 1 and 2 shows the best and average function value for the basic EMD together with one or more of the three advanced features respectively.

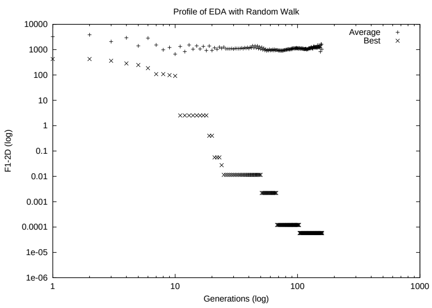
The basic EMD is able to a find a 'good enough' solution, which has a value less than 1, in just 10 generations (see figure 1(a)). This is a period when the diversity in terms of the variety in the parameter values is at the highest. The population converges gradually during this period. As the diversity decreases, EMD tends to try out small variations to the candidate solutions in the population and attempts to update the probability matrix. As the population moves together closer to the optimal solution, a fixed step size will gradually move the candidate solutions across the optimal point and over to the other size of the



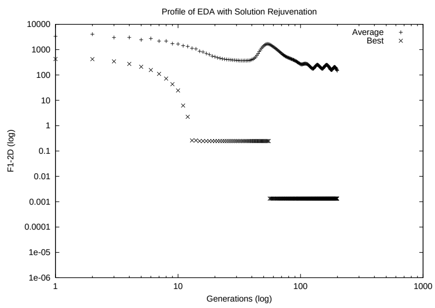
(a) Basic EMD



(b) with adaptive stepsize

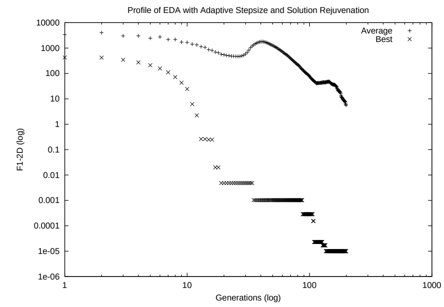


(c) with random walk

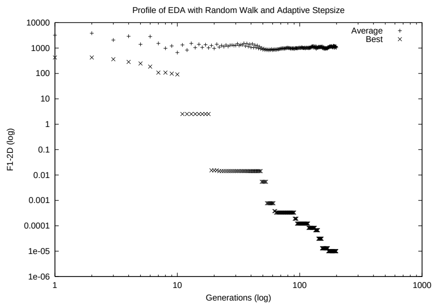


(d) with solution rejuvenation

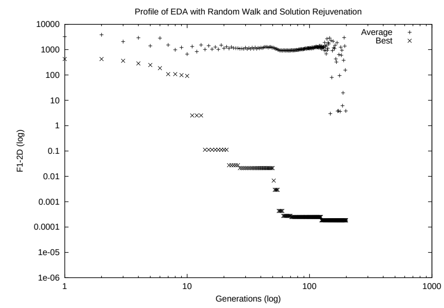
Figure 1: Development of a solution on F1 using basic EMD and EMD with one advanced feature



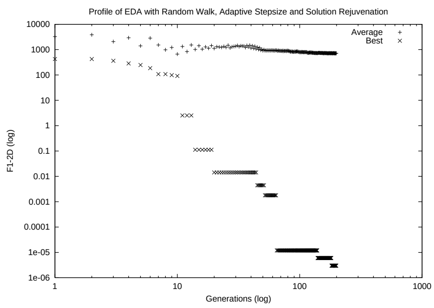
(a) with adaptive stepsize and solution rejuvenation



(b) with random walk and adaptive stepsize



(c) with random walk and solution rejuvenation



(d) with random walk, adaptive stepsize and solution rejuvenation

Figure 2: Profile of the development of a solution on F1 using more than one additional feature

‘valley’ in the function value landscape. In combination with the fact that EMD accepts all new points into the solution pool, the population average will diverge at certain point. In addition, whatever direction biasing information learned in the probability matrix will suddenly become irrelevant. A new ‘learning’ phase then commences (between generations 40-50). This cycle of learning and re-learning can also be observed at later stage of the search (between generations 100 and 200).

Another point worth noting is that new best solution is not discovered at every generation. One might expect that once a trend has been learning and encoded into the probability matrix, any new agent having the latest probability matrix will simply slide downwards towards to the optimal solution. This is not found in any of the plots in figure 1. This is because new candidate solutions are generated using the probability matrix and the step size. However, this fixed size is becoming larger and larger relative to the distance between the best solution and the optimal solution. With a fine tuning mechanism such as the adaptive step size, this problem is eliminated (see figure 1(b)). It can also be observed that the adaptive step size feature produces a population that is converging at a faster rate than the basic EMD. This is another evidence of efficiency gain in utilizing the probability matrix.

The random walk feature (figure 1(c)) seems to be making a more progressive improvement on the best solution than the basic EMD. A distinctive feature of random walk is that it produces a non-converging, if not diverging, population. Notice that the tail of the population average graph is diverging and the search finished before the set limit, at 159 generations, as the number of agents has dropped to zero. This is somewhat undesirable as the search has become more unpredictable but this side effect vanishes when random walk is combined with other features.

The solution rejuvenation feature (figure 1(d)) alone performs almost exactly as the basic EMD. It seems this feature is not making significant contribution in the optimization of a unimodal function.

The combination of any two advanced features shows some interesting results when optimizing the two-dimensional F1. Adaptive step size continue to show the best performance and shadowed the contribution of solution rejuvenation (figure 2(a)). A converging population similar to, but slight different from, the one in figure 1(b) is also observed.

The initial stage of the the random walk and adaptive step size combination (figure 2(b)) is very similar to the random walk only search. But the later stage of the search using both features shows a very steady improvement. However, the population is no longer converging and less fluctuation is observed. It can be concluded that step size adaptation helps to exploit existing solutions while random walk is exploring new areas in the solution space. This is the most important findings of this initial testing.

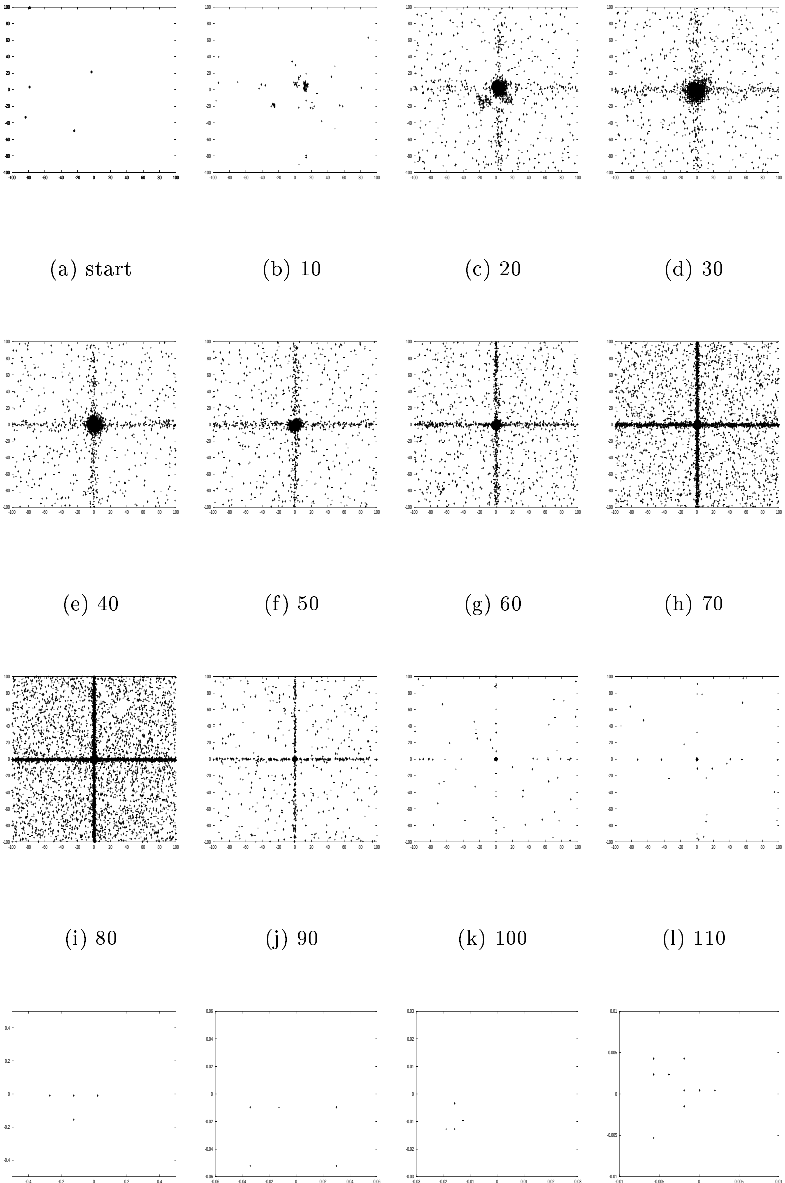
The combination of random walk and solution rejuvenation (figure 2(c)) shows a profile similar to random walk alone. However, the population value at the second half of the search fluctuations instead of simply diverging. This is due to the fact that the few good solutions are not eliminated from the system and allowed to reproduce beyond the time when are freely active agents have become extinct.

When all three advanced features are used, EMD achieves the smallest value and the population shows a slight converging behavior (figure 2(d)). If the function value is chosen as the evaluation criteria of the advanced features when acted alone, adaptive step size achieves the best solution followed by random walk. When more than one feature is considered, all three advanced features together has a slight advantage over adaptive step size alone.

Figure 3 shows the distribution of agents used to optimize the 2 dimensional version of F1. It shows the behavior of the agents from a different perspective. Five agents were positioned randomly on the solution space - three with x-value around -80, one around -20 and one around zero - but none is within the central 20 by 20 zone around the origin. As the search progresses to generation 10, more and more agents appears inside the central zone. The population size increase is due to the success of the agents around there, which *reproduces* quickly; providing some positive feedback to their parents. This trend continues between generation 10 and generation 80. Note that there also are agents throughout the solution space. They were put there by *rational moves* with by step size or by *random moves*.

The diameter of the occupied area increases between generation 10 and 30, and shrinks between generation 30 and 80. The agents learn the 'direction' of the fitness landscape in the first period and zoom in to the origin (the optimal solution) in the second period. The third observation worth mentioning is the increase in the number of agents along the axes, which can be considered suboptimal solutions as one of the two variables is in its lowest value. This is another evidence to show that learning has happened.

The number of agents is at its highest point around generation 80 but dropped drastically in the next 10 generations. This is because agents not around the origin or along the axes have much higher fitness than the popula-



(a) start (b) 10 (c) 20 (d) 30

(e) 40 (f) 50 (g) 60 (h) 70

(i) 80 (j) 90 (k) 100 (l) 110

(m) 120 (n) 160 (o) 200 (p) 220

Figure 3: Distribution of agents in the solution space for 2-D F1

tion average. Therefore, they are removed due to *aging* or *sudden death.* The second phase of the search sees agents along the axes are elimination as they are occupying a suboptimal position. As the agents continues to move towards the origin, agents that are further away from the origin are eliminated. The last four plots in figure 3 zoom in to the origin to show the best agents. In generation 120, four of the five agents are shown (range: ± 0.5). In generation 160, all five agents are inside the ± 0.06 range. In generation 200, four of the six agents are shown (range: ± 0.02). The last generation, 200, has 11 of its 14 agents inside the ± 0.01 range. It is interesting to note that some agents have overshot from the second quadrant to the third quadrant between generation 160 and generation 200. They have adjusted quickly to center around the origin at generation 220 and resulted in an increase in the number of agents.

5 Related Work

Several well-known single-threaded and population-based search algorithms share some common features with the evolutionary multi-agent diffusion (EMD) discussed above and yet they are different. This section attempts to highlight some of their major differences.

5.1 Simulated Annealing

Apart from the difference in population size, both SA and EMD maintains a schedule for modifying the system parameters. In SA, the temperature of the 'cooling' process is predetermined to go down gradually. The counterpart in EMD is the modification to the step size. However, EMD's step size can go up as well as down depending on the progress over a period of time.

Another major difference between SA and EMD lies in the way they accept worse candidate solution as the starting point of the next generation. While SA will accept this candidate based on certain probability, EMD will always accepts it. The aging mechanism in EMD allows the system to get rid of poor performers after they are consistently performing badly over a period of time.

5.2 Evolutionary Algorithms

There are many strands in the class of evolutionary algorithms, which are mostly population-based. However, most of them maintain a fairly common framework based on the 'survival of the fittest' principle. The differences lie in the choice of so called genetic operators and in implementation of the algorithm. In summary, an EA will subject a fix-sized population of individuals to the following operations: selection, reproduction and mutation. These operators are usually applied with certain probabilities but these probabilities remain unchanged during the search. EMD follows a similar procedure but with a few differences.

Firstly, the population size of EA is usually fixed while EMD maintains an open-ended approach. The effect of such an approach allows the weaker agents the chance to survive to the next generation to wander to a better position, and hence free from being trapped by local suboptimal.

Secondly, a variable mutation rate, realized by the diffusion step size, as opposed to a fixed mutation rate allows EMD to migrate smoothly between exploration and exploitation. Moreover, most EA give high priority to making small changes rather than large changes, as realize by a Gaussian distribution. The latter may be able to use fewer time steps for search if larger steps are possible.

Thirdly, selection in EA is a population-wide operation where all individuals in the population are sorted according to fitness. However, selection in EMD is entirely local, as realized in breeding, which compares only the performance of an individual and its parent. This local selection process is computationally less costly and offers better chance for parallel implementation.

Fourthly, the probability matrix in EMD allows individuals to capture the trend of the fitness landscape and points further search towards a potentially fruitful direction. This information is again local to an individual and perhaps only shared by its siblings and its parent. In contrast, any trend in EA can only be captured by observing all individuals in the population, such as the distribution of bit patterns.

Fifthly, the chance to reproduce and remain in the population decreases with the individual's fitness in the case of EA. In contrast, the notion of success is limited to the local sense in EMD. This allows more diversity in the population.

Finally, random walk plays a stronger role in EMD than in EA as an instrument to escape suboptimal.

5.3 Particle Swarm

Particle swarm is to some extent similar to EMD in that individual particles make their own local decisions without consulting any other particles except one. In more details, individual particles requires two pieces of information to make the decision regarding the next position to assume: the velocity (direction and speed) of the last move and that of the overall best particle. Therefore, some kind of global information is always required. Although communication between individual of the same lineage is required in EMD, it does not depend on any global information.

5.4 Ant Colony Optimization

ACO is an optimization algorithm based on the principle of autocatalysis, where shared knowledge is built incrementally and is implicit. Knowledge sharing in EMD is explicit and local, and is achieved via the probability matrix, which is used and updated by all offspring of an agent.

5.5 Cultural Algorithm

Human being passes experience and knowledge to the next generation and beyond by means of culture. This form of knowledge sharing has been captured in the cultural algorithm. EMD also takes the view that knowledge gained by one individual is useful to others. As such, knowledge in the form of direction of change is passed to other via the probability matrix of the parent. This is a low-level knowledge sharing and is consumed by individuals within the 'family'. This is particularly useful in a population-based search algorithm where many sites in the fitness landscape are being explored in parallel. Direction information in one site is usually irrelevant in another.

6 Conclusions

This article has described a new local search algorithm called *evolutionary multi-agent diffusion* (EMD) for global optimization. Analogous to particles diffusing in a fluid by Brownian motion and human migration, a population of agents are used in EMD to represent a set of candidate solutions to the optimization task. There are three distinctive features in EMD.

An agent in EMD makes adjustments to the setting of its object variables using local information only. This information is passed on from its parent in the form of a probability matrix that aims to capture the 'trend' of the fitness landscape in their immediate neighborhood. Moreover, it is shared by this agent's siblings and is updated every time one or more siblings diffuses to another position, no matter it is successful or not. Associated with each agent is set of local behavior, such as diffusion, reproduction, aging and feedback. Communication between agents is restricted to just between parent and its offspring.

EMD can self-adapt the step size used by the agents during diffusion; the only global information that an agent uses. By setting the initial step size to a reasonable value, EMD will increase or decrease its value based on the whether progress has been made or not.

Poor performers are always welcome in EMD. This means they have the right to remain in the population until their lifespan expires. They also have the right to reproduce, once they have reached a solution better than their parents. This strategy helps EMD to main a high diversity in the population.

The initial analysis of the behavior with EMD is that it is able to automatically maintain a good balance between exploration and exploitation. This is exemplified by the continuous discovery of new local best solution while maintaining a substantially high population average. The ability to perform this delicate balancing act also helps EMD to escape from the well-known problem of premature convergence faced by many stochastic search algorithms especially those self-adapting ones [6].

References

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation.* Institute of Physics Publishing and Oxford University Press, 1997.

[2] S. W. Bednarz. Mission geography. http://geog.tamu.edu/sarah/humangeog/migration8.html, 2000.

[3] W. A. V. Clark. *Human Migration.* Sage Publications, 1986.

[4] E. L. Cussler. *Diffusion : Mass Transfer in Fluid Systems.* Cambridge University Press, 1997.

[5] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperative agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26(1):1–13, 1996.

[6] M. Glickman and K. Sycara. Reasons for premature convergence of self-adapting mutation rates. In *Proc. of the 2000 Congress on Evolutionary Computation*, pages 62–69, Piscataway, NJ, 2000. IEEE Service Center.

[7] R. Horst and P. M. Pardalos, editors. *Handbook of Global Optimization.* Kluwer Academic Publishers, 1995.

[8] R. Horst and H. Tuy. *Global Optimization : Deterministic Approaches.* Springer-Verlag, 1990.

[9] J. Kennedy. The particle swarm: Social adaptation of knowledge. In *Proceedings of the 1997 International Conference on Evolutionary Computation*, pages 303–308, Piscataway, NJ, 1997. IEEE Serice Center.

[10] S. Kirkpatrick, C. Gelatt, and M. AndVecchi. Optimization by simulated annealing. *Science*, 220(4598):671 – 680, 1983.

[11] J. Mockus. *Bayesian Approach to Global Optimization : Theory and Applications.* Kluwer Academic, 1989.

[12] R. G. Reynolds. An introduction to cultural algorithms. In A. V. Sebald and L. J. Fogel, editors, *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pages 131–139. World Scientific, River Edge, NJ, 1994.

[13] A. Torn and A. Zilinskas. *Global Optimization.* Sringer-Verlag, 1989.